Notes on polynomially solvable special cases for the unbounded knapsack problem

IIDA Hiroshi

Abstract

This article shows that on two formulations of the unbounded knapsack problem, i.e. maximization and minimization problems, a set of dominance relations for one defines a polynomially solvable special case for the other, and also includes an additional discussion on the special case for the minimization problem.

Keywords: combinatorial optimization; knapsack; dominance relation; polynomially solvable special case; greedy heuristic

1 Introduction

The article deals with the Unbounded Knapsack Problem (UKP, for short). It will be formulated as follows:

$$\begin{aligned} \text{(UKP)} \quad & \text{maximize} \ \sum_{j=1}^{n} c_j x_j \\ & \text{subject to} \ \sum_{j=1}^{n} a_j x_j \leq b \qquad (1) \\ & \qquad x_j \geq 0 \ \text{(integer)}, \ j=1,2,\dots,n. \end{aligned}$$

As different from the ordinary 0–1 knapsack problem of $x_j \in \{0,1\}$, the available number of any item is unbounded as $x_j \geq 0$ where each index j is one-to-one correspondence to an item. Given n types of items (each of which is of profit c_j and weight a_j) and a knapsack of capacity b, the UKP is to pack the items into the knapsack so that the total profit of packed items is maximized without the total weight of those exceeding the capacity. Throughout this article without loss of generality we assume that both c_j and a_j associated with any j-th item and b are all positive integers.

There exist many contributions to UKP in the literature as seen in the book by Nemhauser and Wolsey [5]. For example, a dominance relation is one of them as studied by, e.g. Martello and Toth [4], Dudziński [1], and Zhu and Broughan [6]. Owing to the relation we have a more small-sized and equivalent problem to an instance of UKP given. In addition, a polynomially solvable special case has also been studied by, e.g. Magazine et al [3], Hu and Lenard [2], and Zukerman et al [7].

Before entering the main we briefly mention the contents of this article, which is divided in three: Section 2 connects dominance relations with polynomially solvable special cases; Section 3 gives another proof to the result obtained in the previous section; Section 4 discusses the special case presented in [7] last year.

2 A connection between the two

To take an example of dominance relations, if in an instance of UKP (1) it holds that $a_j \geq a_k$ and $c_j \leq c_k$ then j-th item is said to be dominated by k-th. This means that the optimal value of the instance can be achieved without the j-th since a solution x with $x_j > 0$ could be improved by $x_k \leftarrow x_k + x_j$; $x_j \leftarrow 0$. According to this relation we assume in this section $a_1 < a_2 < \cdots < a_n$ and $c_1 < c_2 < \cdots < c_n$, which are also assumed in [7].

Recently, a condition which defines a polynomially solvable special case for UKP was presented in [7]. The formulation of UKP in [7] is not a maximization (1) but a minimization problem in which the constraint on the capacity is of a form $\geq b$ and its objective function value is minimized. Hereafter we call the problem minimization UKP. The condition presented in [7] is as follows:

$$c_{j+1} \leq \lfloor a_{j+1}/a_j \rfloor c_j, \text{ for } j=1,2,\dots,n-1. \tag{2}$$

In another view, for each j fixed, this is a dominance relation for UKP (1) as in [4], that is, $(j+1)$-st item is dominated.[1]

Conversely the following is, also for each j fixed, a dominance relation for the minimization UKP:

$$c_{j+1} \geq \lceil a_{j+1}/a_j \rceil c_j, \text{ for } j=1,2,\dots,n-1. \tag{3}$$

Here a question arises whether the condition (3) defines a polynomially solvable special case for UKP (1) or not. As to the question the following answers, "Positive."

Observation. For UKP (1) with (3), an optimal solution x^* with $x^*_n = \lfloor b/a_n \rfloor$ exists.

proof. To begin with a trivial case $a_n > b$ may be excluded; otherwise, by (3) there exists an optimal solution x^* in which $x^*_j < \lceil a_{j+1}/a_j \rceil$ for any $j < n$. Indeed, if consider x with $x_k \geq \lceil a_{k+1}/a_k \rceil$ for some $k(<n)$ then y constructed as $y_k = x_k - \lceil a_{k+1}/a_k \rceil$; $y_{k+1} = x_{k+1} + 1$; otherwise $y_j = x_j$ could be improved. Now we negate our goal, which brings $x^*_n < \lfloor b/a_n \rfloor$. Then,

$$\begin{aligned}\sum_{j=1}^{n} c_j x^*_j &\leq \sum_{j=1}^{n-1} (\lceil a_{j+1}/a_j \rceil c_j - c_j) + c_n x^*_n \\ &\leq \sum_{j=1}^{n-1} (c_{j+1} - c_j) + \lfloor b/a_n \rfloor c_n - c_n = \lfloor b/a_n \rfloor c_n - c_1.\end{aligned}$$

On the other hand the negation also brings $\lfloor b/a_n \rfloor c_n < \sum_{j=1}^{n} c_j x^*_j$. Together, we have a contradiction $c_1 < 0$. ∎

[1]By the definition of the floor function we have $a_{j+1} \geq \lfloor a_{j+1}/a_j \rfloor a_j$, for $j=1,2,\dots,n-1$.

Therefore, using (3) recursively we can solve UKP (1) with (3) by an ordinary greedy approach: packing n-th item as many as possible; $b \leftarrow b - \lfloor b/a_n \rfloor a_n$; packing $(n-1)$-st item as many as possible; and so on. A point is that n-th item is the most efficient under (3), that is, (3) implies

$$c_1/a_1 \leq c_2/a_2 \leq \cdots \leq c_n/a_n. \tag{4}$$

3 Another proof

In fact, a necessary and sufficient condition for a case where an ordinary greedy approach solves UKP has already been exposed in [3]. In this section we show that the condition includes (3).

The UKP discussed in [3] is of an equality constraint on the capacity, i.e.

$$z = \min_x \left\{ \sum_{j=1}^{n} c_j x_j \;\middle|\; \sum_{j=1}^{n} a_j x_j = b;\; x_j \geq 0 \text{ (integer)},\; j=1,2,\ldots,n \right\}, \tag{5}$$

where b is a positive integer. Besides the integrality of a_j and c_j, assumptions in [3] are

$$c_1/a_1 \geq c_2/a_2 \geq \cdots \geq c_n/a_n, \tag{6}$$

$$1 = a_1 < a_j,\; 2 \leq j \leq n. \tag{7}$$

Before stating the condition in [3] we introduce two functions: $F_k(y)$ $(1 \leq k \leq n,\ 0 \leq y \leq b)$ as one which is restricted in (5) such that only the first k items are available and a capacity is y (i.e. $z = F_n(b)$); the other $H_k(y)$, under the same restrictions as those on $F_k(y)$, the total profit gained by the greedy approach as in the previous section.

Theorem (Hu and Lenard, 1976) Suppose $H_k(y) = F_k(y)$ for all positive integers y and some fixed k. If $a_{k+1} > a_k$ and p and δ are the unique integers for which $a_{k+1} = pa_k - \delta$ and $0 \leq \delta < a_k$, then the following are equivalent.

(a′) $H_{k+1}(y) \leq H_k(y)$ for all positive integers y,

(a) $H_{k+1}(y) = F_{k+1}(y)$ for all positive integers y,

(b) $H_{k+1}(pa_k) = F_{k+1}(pa_k)$,

(c) $c_{k+1} + H_k(\delta) \leq pc_k$.

Hu and Lenard [2] added (a′) to the one in [3], which simplified the proof. Note that the relation between (a′) and (c) is the same as that between (a) and (b), because (c) is transformed into $H_{k+1}(pa_k) \leq H_k(pa_k)$.[2]

On UKP (1) it is easily shown that Theorem still holds, provided: with (4) instead of (6) since n-th item should be the most efficient for a greedy approach regardless of the formulation of UKP; without (7), which is not used in the proof, as seen in the statement of Theorem however it is assumed that $a_j > 0$ for any j (Hereafter such is denoted as $a > 0$ concisely). A difference is that both '$\leq$' appeared in (a′) and (c) are replaced with '$\geq$' due to being the maximization problem. For the same reason, while there is no restriction on the sign of c_j's in [3] (also in [2]) we shall have $c \geq 0$ on UKP (1). To be more specific, considering $k = 1$ and $c_1 < 0$ we have $H_1(a_1) = c_1 < 0 = F_1(a_1)$. For $H_1 = F_1$ thus $c_1 \geq 0$ is necessary; with (4), it moreover brings $c \geq 0$.[3] Now obvious that $H_k \geq 0$, which validates (3)$\Rightarrow$(c). Consequently it can directly be proved by Theorem that the greedy approach solves UKP (1) with (3) .

4 The special case for the minimization UKP

In this section we focus on the minimization UKP, and assume that $a_1 < a_2 < \cdots < a_n$ and $c_1 < c_2 < \cdots < c_n$ as in Section 2, that is, follow [7].

The work in [7] could roughly be summarized as follows: The condition

---

[2] $H_{k+1}(pa_k) = H_{k+1}(a_{k+1} + \delta) = c_{k+1} + H_{k+1}(\delta) = c_{k+1} + H_k(\delta)$, since $\delta < (a_k <)\, a_{k+1}$.

[3] In addition, using the assumption of Theorem recursively we have $0 < a_1 < a_2 < \cdots < a_n$, which implies $0 \leq c_1 \leq c_2 \leq \cdots \leq c_n$.

(2) implies the existence of an optimal solution x with $x_n \geq \lfloor b/a_n \rfloor$. Then, using (2) recursively, an algorithm proposed in [7] first produces (possible) two solutions such that $x_n = \lceil b/a_n \rceil$ and $x_n = \lfloor b/a_n \rfloor$. The feasible former is a candidate for optimality as is; after $b \leftarrow b - \lfloor b/a_n \rfloor a_n$, based on the infeasible latter in duplicate, it also produces (possible) two so that $x_{n-1} = \lceil b/a_{n-1} \rceil$ is added to one and $x_{n-1} = \lfloor b/a_{n-1} \rfloor$ to the other, respectively; and so on. Last, among at most n candidates produced, the algorithm picks up the best.

The polynomially solvable special case for UKP (1) defined by (3) is that a greedy approach solves it while the case by (2) is not so, which will come from being the minimization problem. Namely, in the minimization UKP an infeasible solution can be made feasible by packing more items whereas in the other two UKPs, once the total weight of a solution has exceeded the capacity, the solution will remain infeasible even by doing so.

In fact, the condition (2) is a sufficient but not a necessary condition for the special case solvable by the algorithm above, which is illustrated by, e.g. an instance of UKP below ($n=2$):

i	1	2
a_i	2	3
c_i	2	4
b	5	

In this, $c_2 > 2 = \lfloor a_2/a_1 \rfloor c_1$; still, the algorithm finds $x_1 = x_2 = 1$. Furthermore the instance does not follow (6) implied by (2). Namely, the algorithm solves the instance even though it does not firstly take account of the most efficient 1st item.

Anyway, which condition is necessary and sufficient to strictly define a class of the minimization UKP solvable by the algorithm above. Here we would like to add that the algorithm fails on the above instance replaced with $b=4$ or 6. To take a wild guess, the one will include the capacity. Especially $n=2$, it could naturally be stated as follows:

$$\min\{\lceil b/a_2\rceil c_2, \lceil (b-ma_2)/a_1\rceil c_1+mc_2\} \leq \min_{0\leq j<m}\{\lceil (b-ja_2)/a_1\rceil c_1+jc_2\},$$

where $m = \lfloor b/a_2 \rfloor$.[4] How about in general? To the best of my knowledge it is still open.

References

[1] K. Dudziński, "A note on dominance relation in unbounded knapsack problems," *Operations Research Letters* **10**(7) 417–419 (1991).

[2] T. C. Hu and M. L. Lenard, "Optimality of a heuristic solution for a class of knapsack problems," *Operations Research* **24**(1) 193–196 (1976).

[3] M. J. Magazine, G. L. Nemhauser, and L. E. Trotter, Jr., "When the greedy solution solves a class of knapsack problems," *Operations Research* **23**(2) 207–217 (1975).

[4] S. Martello and P. Toth, "An exact algorithm for large unbounded knapsack problems," *Operations Research Letters* **9**(1) 15–20 (1990).

[5] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, 1999 (paperback).

[6] N. Zhu and K. Broughan, "On dominated terms in the general knapsack problem," *Operations Research Letters* **21**(1) 31–37 (1997).

[7] M. Zukerman, L. Jia, T. Neame, and G. J. Woeginger, "A polynomially solvable special case of the unbounded knapsack problem," *Operations Research Letters* **29**(1) 13–16 (2001).

---

[4] In the case where $m = 0$, i.e. $a_2 > b$, the right hand side of the inequality is empty nonetheless we may assume that it holds even in that case.